Graph500 benchmark with automatic performance tuning

Masahiro Nakao
RIKEN R-CCS
Japan
masahiro.nakao@riken.jp

Koji Ueno
Fixstars Corporation
Japan
koji.ueno@fixstars.com

Katsuki Fujisawa
Kyushu University
Japan
fujisawa@imi.kyushu-u.ac.jp

Yuetsu Kodama
RIKEN R-CCS
Japan
yuetsu.kodama@riken.jp

Mitsuhisa Sato
RIKEN R-CCS
Japan
msato@riken.jp

1 INTRODUCTION

In various fields such as social networks and drug discovery, there are many attempts to represent data relationships as graph structures and analyze them on computers at high speed. We have been developing breadth-first search (BFS) on the Graph500 benchmark applying various techniques[1], and have achieved the world's top performance in the Graph500 list (https://graph500.org) as of the time of writing (October 2023). However, since most existing research, including our study, targets specific graphs and computer systems, the burden of performance tuning for users has become an issue. Therefore, this study develops an automatic performance tuning function that automatically determines the optimal parameters for BFS in the Graph500 benchmark.

2 HYBRID BREADTH-FIRST SEARCH

In the Graph500 list, Hybrid-BFS algorithm[2], shown in Fig. 1, is often used. In Hybrid-BFS, BFS proceeds by switching between the conventional top-down search method **A** and another search method called bottom-up search **B**. The current starting points are ②, and unexplored adjacent points ◯ are searched. One issue with top-down search is that the current starting points ② need to check all the adjacent points, but since most of the adjacent points have already been searched (the first starting point ① and the current starting points ② have already been searched), many redundant checks occur in the middle stage of BFS. The arrow represents the check. In bottom-up search, the current starting points ② are searched from the unexplored vertices ◯, which is the opposite direction of top-down search. The advantage of bottom-up search is that it reduces redundant checks because the check can be interrupted if even a single current starting point ② is found.

3 AUTOMATIC PERFORMANCE TUNING

The Hybrid-BFS process starts top-down, goes bottom-up in the middle, and ends top-down. The timing of these two transitions has a strong influence on performance. The paper[2] recommends $m_f > m_u/\alpha$ for the switch from top-down to bottom-up and $n_f < n/\beta$ for the switch from bottom-up to top-down. The m_f is the number of edges of the vertex being searched, the m is the number of edges in the graph, the n_f is the number of vertices being searched, and the n is the number of vertices in the graph. The α and β are constant parameters that can be determined by users.

We propose an algorithm to automatically determine α and β. Graph500 benchmark performs BFS using 64 different vertices in a certain graph as starting points. Our algorithm records the timing of the switch between top-down and bottom-up, and executes the following flow. (1) For each execution, calculate α', which can be switched at a different timing. (2) Re-run the execution with α' closest to α. (3) If the performance has improved, $\alpha' \rightarrow \alpha$, and return to (2). (4) Perform the same operations as (1)-(3) for β.

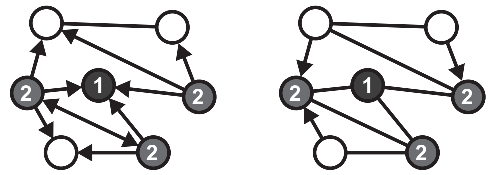

Figure 1: Breadth-first search

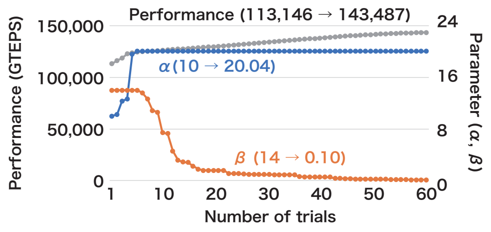

Figure 2: Performance result

4 RESULT

Fig. 2 shows the result using 152,064 nodes of the supercomputer Fugaku. The initial values of α and β were set to 10 and 14, respectively. Finally, the values changed to 20.04 and 0.10, respectively, and the performance achieved about 27% performance improvement from 113,146 GTEPS to 143,487 GTEPS.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number 21H03450.

REFERENCES

[1] Masahiro Nakao et al. Performance of the Supercomputer Fugaku for Breadth-First Search in Graph500 Benchmark. ISC 2021
[2] Scott Beamer et. al. Distributed Memory Breadth-First Search Revisited: Enabling Bottom-Up Search, IPDPSW 2013